**Mocking**

### **Bonussystem für die Boniberechnungen von Mitarbeitern**

Im Rahmen dieses Projekts soll ein flexibles und erweiterbares Bonussystem für Mitarbeiterbewertungen entwickelt werden. Das Ziel ist es, den jährlichen Bonus eines Mitarbeiters anhand mehrerer Einflussfaktoren zu berechnen. Zu beachten ist dabei zunächst, dass die Firma zwei verschiedene Arten von Boni kennt: a) einen Grundbonus und b) einen Performance-Bonus, der den Grundbonus anhand der individuellen Leistungsbewertung (Performance) eines Mitarbeiters anpasst. Mitarbeiter werden durch ihren Namen, ihrer Firmenzugehörigkeit (in Jahren), ihrer Performance-Bewertung (in Punkten), der Anzahl abgeschlossener Projekte und Fehltage beschrieben. Der Se**niority-Bonus** erhöht den Bonus basierend auf der Betriebszugehörigkeit des Mitarbeiters, indem in bestimmten Intervallen (zum Beispiel alle 5 Jahre 150 Euro mehr pro Jahr) ein fester Betrag hinzugefügt wird. Der **Project-Completion-Bonus** erhöht den Bonus abhängig von der Anzahl erfolgreich abgeschlossener Projekte. Teamleader erhalten ebenfalls einen Bonus. Auch für besonders geringe Fehlzeiten gibt es einen Bonus. Dieser Bonus soll zwei Stufen haben. Anhand der Performance-Bewertung wird der errechnete Bonus mit einem bestimmten Faktor (drei Abstufungen) multipliziert. Für besonders viele Fehltage gibt es Bonusabzüge (zwei Stufen). Die Klasse BonusCalculator benutzt als Client dieses mittels Decorator-Pattern zu implementierenden Bonus-Systems, indem sie die Gesamtberechnung übernimmt. Der BonusCalculator muss dabei noch Unternehmensrestriktionen berücksichtigen, die ein Bonus-Service zur Verfügung stellt. Der Service stellt dabei einen Höchstbetrag, die ein berechneter Gesamtbonus nicht überschreiten darf, sowie einen Mindestbetrag, der nicht unterschritten werden darf, bereit. Diese Restriktionen sollen innerhalb einer Methode zur Anwendung kommen, die einen berechneten Bonus je nach Unternehmensrestriktionen korrigiert, zum Beispiel, indem ein zu hoher Bonus auf den Höchstbetrag korrigiert wird. Denke dir für alle Boni geeignete Beträge und Abstufungen aus.

1. Informiere dich über die Funktionsweise des Decorator-Pattern.
2. Entwickelt ein Klassendiagramm der Gesamtanwendung. Sie besteht aus dem Decorator-Pattern sowie weiteren Klassen! Beschreibt in der Readme eures Projekt das Bonussystem, das ihr euch anhand der Aufgabenstellung ausgedacht habt (bei welchen Abstufen gibt es welche Beträge?). Gebt eurer Klassendiagramm über diesen Auftrag ab.
3. Implementiert eure Planung testgetrieben in Java unter Benutzung von AssertJ und Mockito. Nach jedem Test und zugehöriger Implementierung sowie, wenn nötig, einem Refactoring, ist zu committen. Jeder Red-Green-Refactor-Cyclus soll also in einem Commit abgebildet und versioniert werden. Gebt euer Projekt über einen Gitlink über diesen Auftrag ab. Wenn bestimmte Funktionalität nicht umgesetzt werden konnte, ist dieses in der Readme zu vermerken.